

In the Claims

Please replace all prior versions, and listings, of claims in the application with the following listing of claims, including marked-up claims with insertions indicated by underlining and deletions indicated by strikeouts and/or double bracketing:

Listing of Claims

1. (Original) A method for transmitting Internet Key Exchange (IKE) data packets across a network comprising the steps of:
 - generating and transmitting an IKE packet over a network;
 - determining whether a response to the IKE packet was received;
 - fragmenting the IKE packet into a plurality of smaller packets when a response is not received, wherein each of the smaller packets includes a header formatted according to the IKE protocol; and
 - transmitting each of the plurality of smaller packets over a network.
2. (Original) The method of claim 1 wherein each header includes an identifier that maybe used to associate the smaller packet with a corresponding IKE packet.
3. (Currently amended) A network node that communicates with other network nodes according to the Internet Key Exchange (IKE) protocol comprising:
 - a User Datagram Protocol (UDP) stack that is capable of generating UDP data packets for transmission over a network;
 - an IKE protocol stack that generates IKE data packets that are subsequently processed by the UDP protocol stack; and
 - a fragmenter module that intercepts IKE data packets prior to being processed by the UDP protocol stack and splits the IKE data packets into a plurality of smaller data packets that may be subsequently formatted by the UDP protocol stack,
 - wherein, each of the plurality of smaller data packets includes a header formatted according to the IKE protocol.
4. (Original) A method for fragmenting a data packet comprising the steps

of: generating an IKE data packet;
intercepting the IKE data packet before it is passed to a subsequent network protocol stack;
determining a maximum size for fragments of an IKE data packet;
dividing the IKE data packet into at least two smaller packets; and
prepending a header to each smaller packet, wherein each header for each smaller packet includes an identifier that associates the smaller packet with its corresponding IKE data packet.

5. (Original) The method of claim 4 wherein the dividing step is performed such that the combined size of each smaller packet and prepended header will not exceed the maximum size.

6. (Original) A method for receiving fragmented Internet Key Exchange (IKE) data packets comprising the steps of:

receiving a plurality of fragments of an IKE data packet from a transmitting node, wherein each fragment includes an identifier that associates each fragment with an IKE data packet; and

discarding all fragments that contain a first identifier if a predetermined number of fragments are received that contain a second identifier.

7. (Original) The method according to claim 6 wherein the step of discarding all fragments that contain a first identifier is performed when at least one fragment is received that contains a second identifier.

8. (Original) The method according to claim 6 further comprising the steps of determining whether all fragments that are associated with an IKE data packet have been received; and

sending a no acknowledgment (NAK) message to the transmitting node when at least one fragment has not been received.

9. (Original) The method according to claim 6 further comprising the step of determining the total size of all fragments that contain the same identifier and discarding said fragments when the total size exceeds a predetermined limit.
10. (Original) The method according to claim 9 wherein the predetermined limit is 64kilobytes.
11. (Currently amended) A system for transmitting Internet Key Exchange (IKE) protocol data packets across a network comprising:
 - means for generating an IKE packet;
 - means for detecting whether the IKE packet was successfully received at the intended receiver node; and
 - means for fragmenting the IKE packets into smaller packets when the IKE packet was not successfully received at the receiver node, wherein each of the smaller packets[[,]] includes information that permits a receiver node to identify the IKE packet associated with each smaller packet and the position of each smaller packet within the IKE packet.
12. (Original) The system of claim 11 further comprising means for determining the capability of the receiver node for receiving fragmented packets.
13. (Original) A method for transmitting data packets across a network comprising the steps of:
 - generating and transmitting an Internet Key Exchange (IKE) packet over a network; determining whether a response to the IKE packet was received;
 - fragmenting the IKE packet into a plurality of smaller packets when a response is not received; and
 - transmitting each of the plurality of smaller packets over a network.
14. (Previously presented) The method of claim 13 wherein each of the plurality of small packets contains a header formatted according to the IKE protocol.

15. (Previously presented) The method of claim 13 wherein the IKE packet contains a header formatted according to the IKE protocol.

16. (Previously presented) The method of claim 15 wherein the plurality of smaller packets contain the same information as that contained within the original IKE packet.

17. (Previously presented) The method of claim 16 wherein at least one of the plurality of smaller packets contains the header formatted according to the IKE protocol.

18. (Currently amended) A method for transmitting data packets of arranging information for transmission across a network comprising the steps of:

generating a data packet containing Internet Key Exchange (IKE) information;
determining whether fragmentation of the data packet is necessary to
successfully transmit the IKE information over a network; and

fragmenting the data packet if necessary into a plurality of smaller packets that may be transmitted over a network,

wherein the steps of generating, determining and fragmenting are performed independently of performing any steps on the data packet corresponding to a transport layer protocol and/or a network layer protocol.

19. (Previously presented) The method of claim 18 wherein the step of determining whether fragmentation is necessary is not based exclusively on the size of the data packet.

20. (Currently amended) A method for resolving transmitting errors associated with transmitting Internet Key Exchange (IKE) packets via protocol stacks that implement the Transmission Control Protocol (TCP), the User Datagram Protocol (UDP), and/or the Internet Protocol (IP) comprising the steps of:

generating a data packet containing IKE data;
fragmenting the packet ~~with~~into a plurality of fragments using a code module that does not implement the TCP, UDP or IP protocols before the packet is processed by a code module

that does implement the TCP, UDP or IP protocols, comprising including an identifier that identifies the data packet in each packet fragment; and
transmitting the ~~fragmented~~ packet fragments over a network.

21. (Previously presented) The method of claim 20 further including the step of determining whether it is necessary to fragment IKE data packets before fragmenting the IKE data packet.

22. (Previously presented) A method for intelligently discarding data packets to efficiently manage resources comprising:

receiving a plurality of data packets containing Internet Key Exchange (IKE) information, wherein the packets were transmitted from a transmitting node in a order that can be determined from information contained within the received data packets;

determining from information contained within the received data packets whether any of the received packets have been received in an order that differs from the order in which the packets were transmitted from the transmitting node; and

discarding at least certain of the received packets when a predetermined number of out of order packets have been received.

23. (Previously presented) The method of claim 22 further including the step of sending a message to the transmitting node that out of order packets have been received.